Make livecd ISO

Jump to bottom

Emmett1 edited this page on Jun 16, 2020 · 11 revisions

Introduction

In this guide i'm targetting LFS but should work on other distro too. Your LFS system that need to make livecd can't be running, so get any distro livecd to boot from, or do it from your currently running distro.

Host requirement

- squashfs-tools
- wget/curl (for downloading syslinux source)
- git (to clone this mkinitrd repository)
- libisoburn (for generating the iso)

Target requirement

- cpio/libarchive
- installed kernel
- squashfs-tools (if you plan to use this created iso to install into hdd)

Target system kernel config requirement (not 100% sure, need tester)

- CONFIG_OVERLAY_FS=m
- CONFIG_CDROM=m
- CONFIG_SQUASHFS=m

Work directory structure

```
live/
|- boot/
|   |- vmlinuz              <- kernel for iso to boot taken from target system
|   |- initrd               <- initrd generated inside chrooted target system
|   \- filesystem.sfs       <- squashed target system
|- isolinux/
|   |- ...
|   |- ...                  <- all syslinux stuffs that copied over below
|   \- ...
\- rootfs/                  <- this directory act like root (/) in live environment
    |- etc/
    |   \- fstab            <- blank fstab file created below
    |- root/
    |   \- custom_script.sh  <- shell script executed right before entering live
environment
    \- ...                  <- you can put your other stuffs here
```

Now time to build the ISO

I'm assuming your LFS system is in `/mnt/lfs` directory. Command in this guide that start with `$` is running as normal user and `#` is running as root.

Cloning mkinitrd repository

```
$ git clone https://github.com/emmett1/mkinitrd
```

Fetch latest syslinux source

```
$ wget -c http://www.kernel.org/pub/linux/utils/boot/syslinux/syslinux-6.03.tar.xz
```

Prepare working directory for livecd

- `isolinux` directory contains syslinux files to boot.
- `boot` directory contains kernel, initrd and squashed system.
- `rootfs` directory that act like root (/) in live environment.

```
$ mkdir -p live/{isolinux,boot}
```

Extract syslinux source then copy required files

```
$ tar xvf syslinux-6.03.tar.xz
$ cp syslinux-6.03/bios/com32/elflink/ldlinux/ldlinux.c32 live/isolinux
$ cp syslinux-6.03/bios/com32/chain/chain.c32 live/isolinux
$ cp syslinux-6.03/bios/core/isolinux.bin live/isolinux
$ cp syslinux-6.03/bios/com32/libutil/libutil.c32 live/isolinux
$ cp syslinux-6.03/bios/com32/modules/reboot.c32 live/isolinux
$ cp syslinux-6.03/bios/com32/menu/menu.c32 live/isolinux
$ cp syslinux-6.03/bios/com32/lib/libcom32.c32 live/isolinux
$ cp syslinux-6.03/bios/com32/modules/poweroff.c32 live/isolinux
```

Squash your LFS system

I'm using `zstd` compression for squashing filesystem, you can use other compression like `xz` compression if zstd not supported on your host's squashfs-tools. You can use `-e <directory>` to exclude directory into your squashed filesystem. Here i'm excluding /dev, /proc, /sys, /run, /root and /tmp directory. I'm sure you dont want to include junks in your livecd iso.

```
# mksquashfs /mnt/lfs live/boot/filesystem.sfs \
    -b 1048576 \
    -comp zstd \
    -e /mnt/lfs/root/* \
    -e /mnt/lfs/tools* \
    -e /mnt/lfs/tmp/* \
    -e /mnt/lfs/dev/* \
    -e /mnt/lfs/proc/* \
    -e /mnt/lfs/sys/* \
    -e /mnt/lfs/run/*
```

Generate initrd using mkinitrd script cloned before

First you need to install `mkinitrd` into target system by using DESTDIR

```
# make -C mkinitrd DESTDIR=/mnt/lfs install
```

Then chroot into target system. Use chroot method on chapter 6.81 on LFS book. Make sure you follow chapter 6.2 first to prepare virtual kernel file system before entering chroot environment. Then generate initrd by running `mkinitrd -k <kernelversion> -a livecd -o /boot/initrd-<kernelversion>.img`. `-k kernelversion` is required if you using different kernel version between host and target system, `-a livecd` is required to execute livecd hook. `initrd` will be created into `/boot` directory with name `initrd-<kernelversion>.img`. Make sure initrd is generated then exit chroot environment. Dont forget to unmount virtual kernel filesystem (dev, proc, sys and run). If you want to change livecd label, modify it in `/etc/mkinitrd.d/livecd.run` in target system and on xorriso below. Default label is `LIVECD`.

```
(chroot) # mkinitrd -k <kernelversion> -a livecd -o /boot/initrd-<kernelversion>.img
(chroot) # exit
```

Copy kernel and initrd into live directory

I'm assume your installed kernel with name `vmlinuz-<kernelversion>`. So copy over both kernel and initrd with name `vmlinuz` and `initrd` respectively into `live/boot/` directory.

```
$ cp -v /mnt/lfs/boot/vmlinuz-<kernelversion> live/boot/vmlinuz
$ cp -v /mnt/lfs/boot/initrd-<kernelversion>.img live/boot/initrd
```

Write isolinux.cfg file

Write isolinux.cfg file into `live/isolinux/` directory. Heres is my own simple isolinux.cfg file.

```
$ cat live/isolinux/isolinux.cfg
UI /isolinux/menu.c32
DEFAULT silent
TIMEOUT 100

MENU VSHIFT 3

LABEL silent
        MENU LABEL Boot MyOwn Linux
        KERNEL /boot/vmlinuz
        APPEND initrd=/boot/initrd quiet

LABEL debug
        MENU LABEL MyOwn Linux (Debug)
        KERNEL /boot/vmlinuz
        APPEND initrd=/boot/initrd verbose
```

```
LABEL existing
        MENU LABEL Boot existing OS
        COM32 chain.c32
        APPEND hd0 0

LABEL reboot
        MENU LABEL Reboot
        COM32 reboot.c32

LABEL poweroff
        MENU LABEL Poweroff
        COM32 poweroff.c32
```

Optional stuffs

You can put files into the iso which not in the squashed system, but appeared in live environment. Example like `/etc/issue`, `/etc/fstab` and etc. Create `rootfs` directory inside live directory to put these stuffs. In this guide i'm making blank fstab file (recommended) so when booting into live environment it wont intefere with current fstab in squashed system.

```
# mkdir -p live/rootfs/etc
# echo "# blank fstab" > live/rootfs/etc/fstab
```

One more thing you can write a shell script to be execute right before booting live environment. This is good if your want enable service, set locale, username, password and etc but wont touch squashed system. This script should located at `live/rootfs/root/custom_script.sh`. This script executed using `chroot`, so `/` in the script mean `/` in live environment.

Finally, generate the iso

```
# xorriso -as mkisofs \
    -isohybrid-mbr syslinux-6.03/bios/mbr/isohdpfx.bin \
    -c isolinux/boot.cat \
    -b isolinux/isolinux.bin \
    -no-emul-boot \
    -boot-load-size 4 \
    -boot-info-table \
    -eltorito-alt-boot \
    -no-emul-boot \
    -isohybrid-gpt-basdat \
    -volid LIVECD \
    -o MyOwnLinuxLiveCD.iso live
```

The iso should be created into current directory with name `MyOwnLinuxLiveCD.iso` .

Disclaimer